IoT Network Architecture and Design

- Imagine that one day you decide to build a house
- To successfully complete a construction project, time and effort are required to design each phase, from the foundation to the roof.
- Your plans must include detailed designs for the electrical, plumbing, heating, and security systems

- A computer network should be built with-- careful planning, security policies, and adherence to well-understood design practices.
- Failure to meet these will likely result in something that is difficult to scale, manage, adapt to organizational changes, and, worst of all, troubleshoot when things go wrong
- If the network fails, company operations can be seriously impaired

- Just as a house must be designed with the strength to withstand potential natural disasters, such as seismic events and hurricanes,
- information technology (IT) systems need to be designed to withstand "network earthquakes," such as
  - distributed denial of service (DDoS) attacks,
  - future growth requirements,
  - network outages, and
  - even human error

DRIVERS BEHIND NEW NETWORK ARCHITECTURES

- Building residential houses vs building a massive stadium..
- The difference between IT and IoT networks is much like the difference between residential architecture and stadium architecture
- The key difference between IT and IoT is the data.
  - IT systems are mostly concerned with reliable and continuous support of business applications such as email, web, databases, CRM systems, and so on.
  - IoT is all about the data generated by sensors and how that data is used.

- The essence of IoT architectures thus involves how the data is transported, collected, analyzed, and ultimately acted upon

Challenge	Description	IoT Architectural Change Required
Scale	The massive scale of IoT endpoints (sensors) is far beyond that of typical IT networks.	The IPv4 address space has reached exhaustion and is unable to meet IoT's scalability requirements. Scale can be met only by using IPv6. IT networks continue to use IPv4 through features like Network Address Translation (NAT).
Security	IoT devices, especially those on wireless sensor networks (WSNs), are often physically exposed to the world.	Security is required at every level of the IoT network. Every IoT endpoint node on the network must be part of the overall security strategy and must support device-level authentication and link encryption. It must also be easy to deploy with some type of a zero-touch deployment model.
Devices and networks constrained by power, CPU, memory, and link speed	Due to the massive scale and longer distances, the networks are often constrained, lossy, and capable of supporting only minimal data rates (tens of bps to hundreds of Kbps).	New last-mile wireless technologies are needed to support constrained IoT devices over long distances. The network is also constrained, meaning modifications need to be made to traditional network-layer transport mechanisms.

The massive volume of data generated	The sensors generate a massive amount of data on a daily basis, causing network bottlenecks and slow analytics in the cloud.	Data analytics capabilities need to be distributed throughout the IoT network, from the edge to the cloud. In traditional IT networks, analytics and applications typically run only in the cloud.
Support for legacy devices	An IoT network often comprises a collection of modern, IP-capable endpoints as well as legacy, non-IP devices that rely on serial or proprietary protocols.	Digital transformation is a long process that may take many years, and IoT networks need to support protocol translation and/or tunneling mechanisms to support legacy protocols over standards-based protocols, such as Ethernet and IP.
The need for data to be analyzed in real time	Whereas traditional IT networks perform scheduled batch processing of data, IoT data needs to be analyzed and responded to in real-time.	Analytics software needs to be positioned closer to the edge and should support real-time streaming analytics. Traditional IT analytics software (such as relational databases or even Hadoop), are better suited to batch-level analytics that occur after the fact.

Security

- IT networks use firewall, IT endpoints are behind firewall
- IoT endpoints are often located in wireless sensor networks that use unlicensed spectrum and are not only visible to the world through a spectrum analyzer but often physically accessible
- IoT systems require consistent mechanisms of authentication, encryption, and intrusion prevention techniques

- IoT systems must:
  - Be able to identify and authenticate all entities involved in the IoT service (that is, gateways, endpoint devices, home networks, roaming networks, service platforms)
  - Ensure that all user data shared between the endpoint device and back-end applications is encrypted
  - Comply with local data protection legislation so that all data is protected and stored correctly
  - Take network-level approach to security in addition to device level approach

Constrained Devices and Networks

- Most IoT sensors are designed for a single job, and they are typically small and inexpensive
- They often have limited power, CPU, and memory, and they transmit only when there is something important
- The networks that provide connectivity also tend to be very lossy and support very low data rates.
- This is a completely different situation from IT networks
- IoT requires a new breed of connectivity technologies that meet both the scale and constraint limitations

Legacy Device Support

- Supporting legacy devices in an IT organization is not usually a big problem
- If someone's computer or operating system is outdated, she simply upgrades
- If someone is using a mobile device with an outdated Wi-Fi standard, such as 802.11b or 802.11g, you can simply deny him access to the wireless network, and he will be forced to upgrade

- In OT systems, end devices are likely to be on the network for a very long time—sometimes decades.
- As IoT networks are deployed, they need to support the older devices already present on the network, as well as devices with new capabilities

COMPARING IOT ARCHITECTURES

- oneM2M architecture
- The IoTWorld Forum(IoTWF)

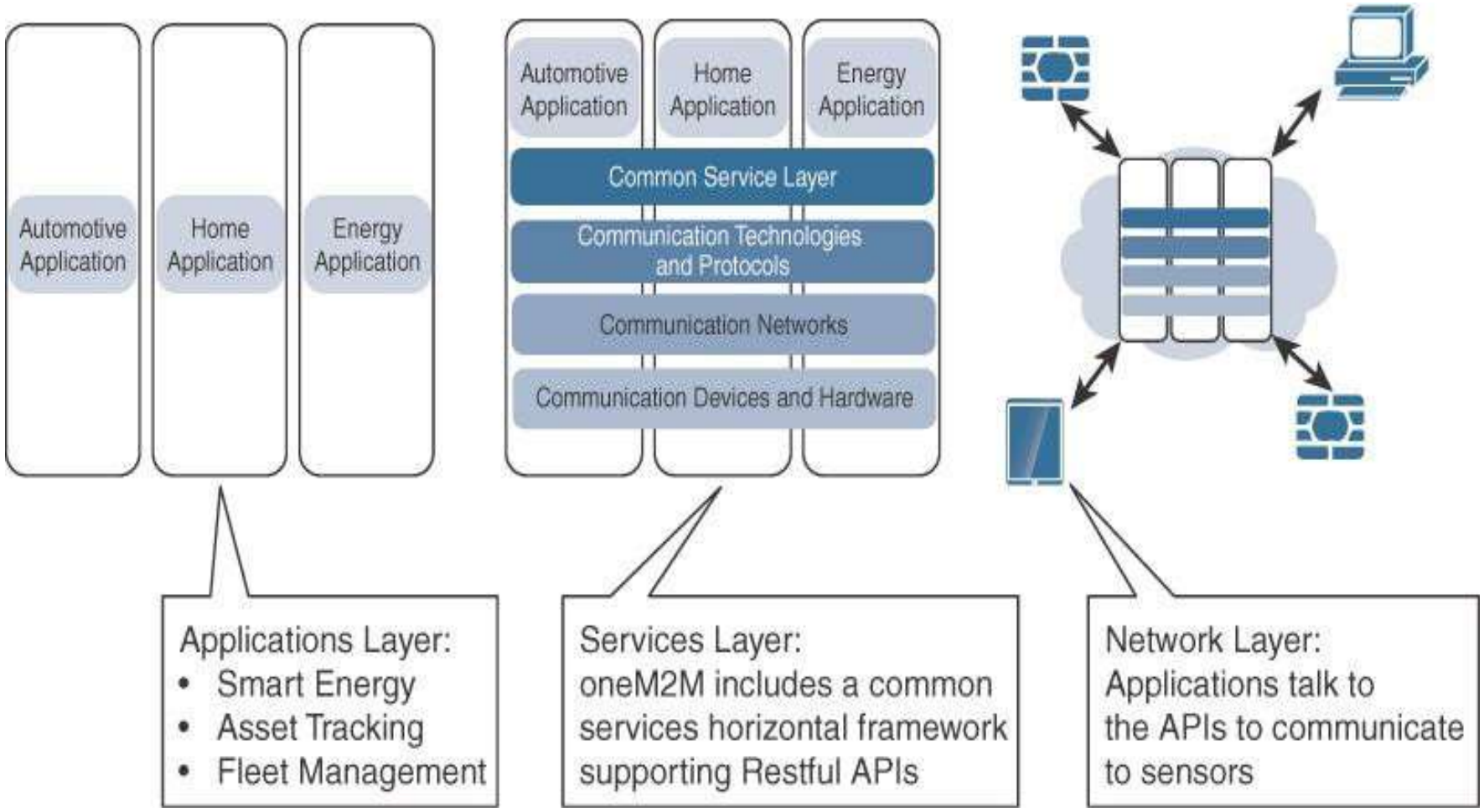
The oneM2M IoT Standardized Architecture

- To standardize the rapidly growing field of machine-to-machine (M2M) communications, the European Telecommunications Standards Institute (ETSI) created the M2MTechnical Committee in 2008
- The goal of oneM2M is to create a common services layer, which can be readily embedded in field devices to allow communication with application servers
- oneM2M's framework focuses on IoT services, applications, and platforms

- One of the greatest challenges in designing an IoT architecture is dealing with the heterogeneity of devices, software, and access methods
- By developing a horizontal platform architecture, oneM2M is developing standards that allow interoperability at all levels of the IoT stack
- For example, you might want to automate your HVAC system by connecting it with wireless temperature sensors spread throughout your office

- The problem is that the LoRaWAN network and the BACnet system that your HVAC and BMS run on are completely different systems and have no natural connection point
- This is where the oneM2M common services architecture comes in.
- oneM2M's horizontal framework and RESTful APIs allow the LoRaWAN system to interface with the building management system over an IoT network, thus promoting end-to-end IoT communications in a consistent way, no matter how heterogeneous the networks

The Main Elements of the oneM2M IoT Architecture



- Three major domains:
- The application layer,
- The services layer, and
- The network layer

Application layer

- The oneM2M architecture gives major attention to connectivity between devices and their applications
- This domain includes the application-layer protocols and attempts to standardize northbound API definitions for interaction with business intelligence (BI) systems

Services layer

- This layer is shown as a horizontal framework across the vertical industry applications
- At this layer, horizontal modules include the physical network that the IoT applications run on, the underlying management protocols, and the hardware.
- Examples include backhaul communications via cellular, MPLS networks, VPNs, and so on.
- Riding on top is the common services layer.
- This conceptual layer adds APIs and middleware supporting third-party services and applications

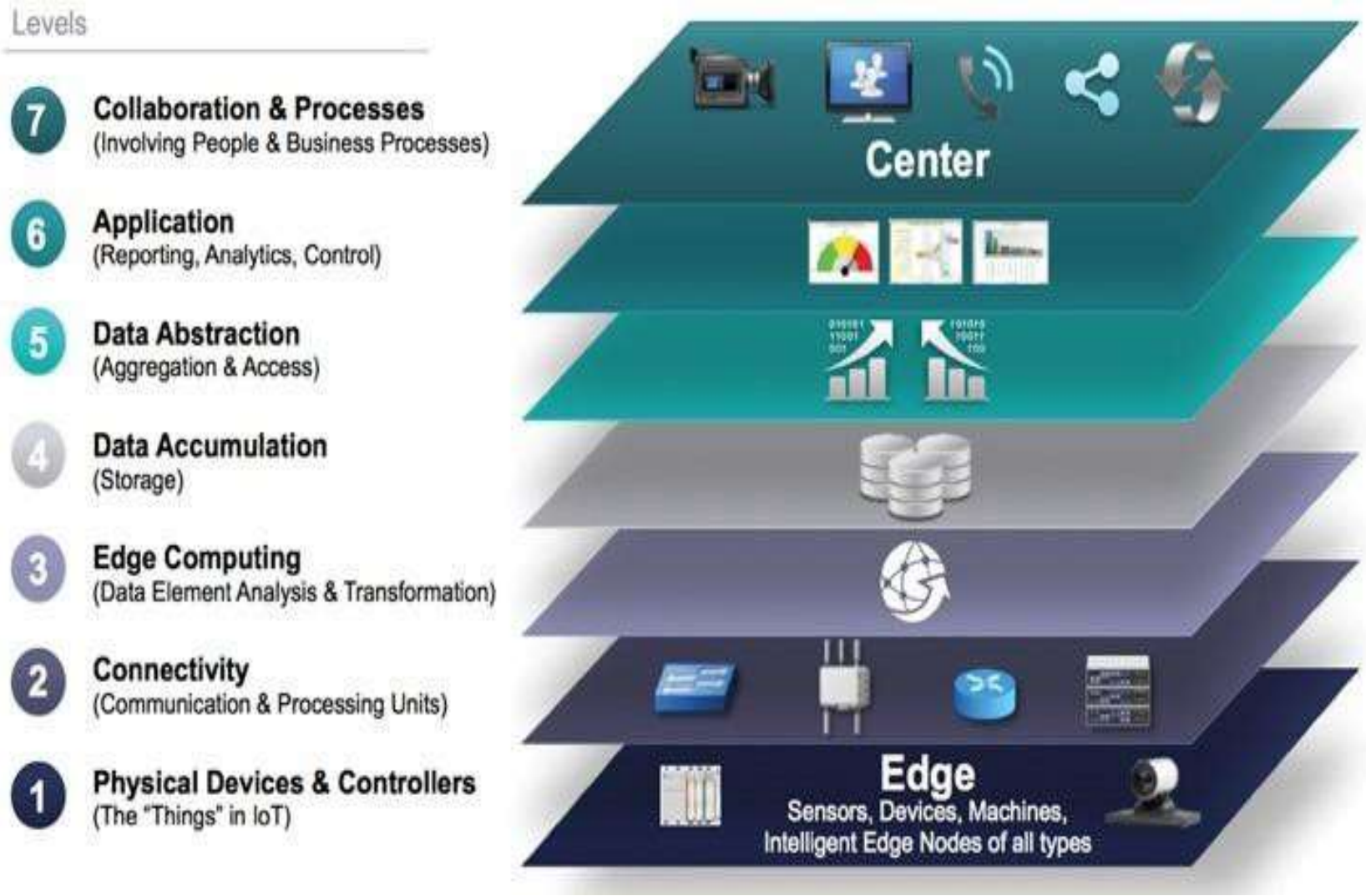
Network layer

- This is the communication domain for the IoT devices and endpoints.
- It includes the devices themselves and the communications network that links them
- Communications infrastructure include wireless mesh technologies, such as IEEE 802.15.4, and wireless point-to-multipoint systems, such as IEEE 801.11ah

The IoT World Forum (IoTWF) Standardized Architecture

- A seven-layer IoT architectural reference model published by IoTWF architectural committee (Cisco, IBM, Rockwell Automation)
- Edge computing
- Data storage
- Access

IoT Reference Model Published by the IoT World Forum



- Using this reference model, we are able to achieve the following:
- Decompose the IoT problem into smaller parts
- Identify different technologies at each layer and how they relate to one another
- Define a system in which different parts can be provided by different vendors
- Have a process of defining interfaces that leads to interoperability
- Define a tiered security model that is enforced at the transition points between levels

Layer 1: Physical Devices and Controllers Layer

- The various endpoint devices and sensors that send and receive information
- The size of these "things" can range from almost microscopic sensors to giant machines in a factory.
- Their primary function is generating data and being capable of being queried and/or controlled over a network.

Layer 2: Connectivity Layer

- The most important function of this IoT layer is the reliable and timely transmission of data.
- More specifically, this includes transmissions between Layer 1 devices and the network and between the network and information processing that occurs at Layer 3 (the edge computing layer).

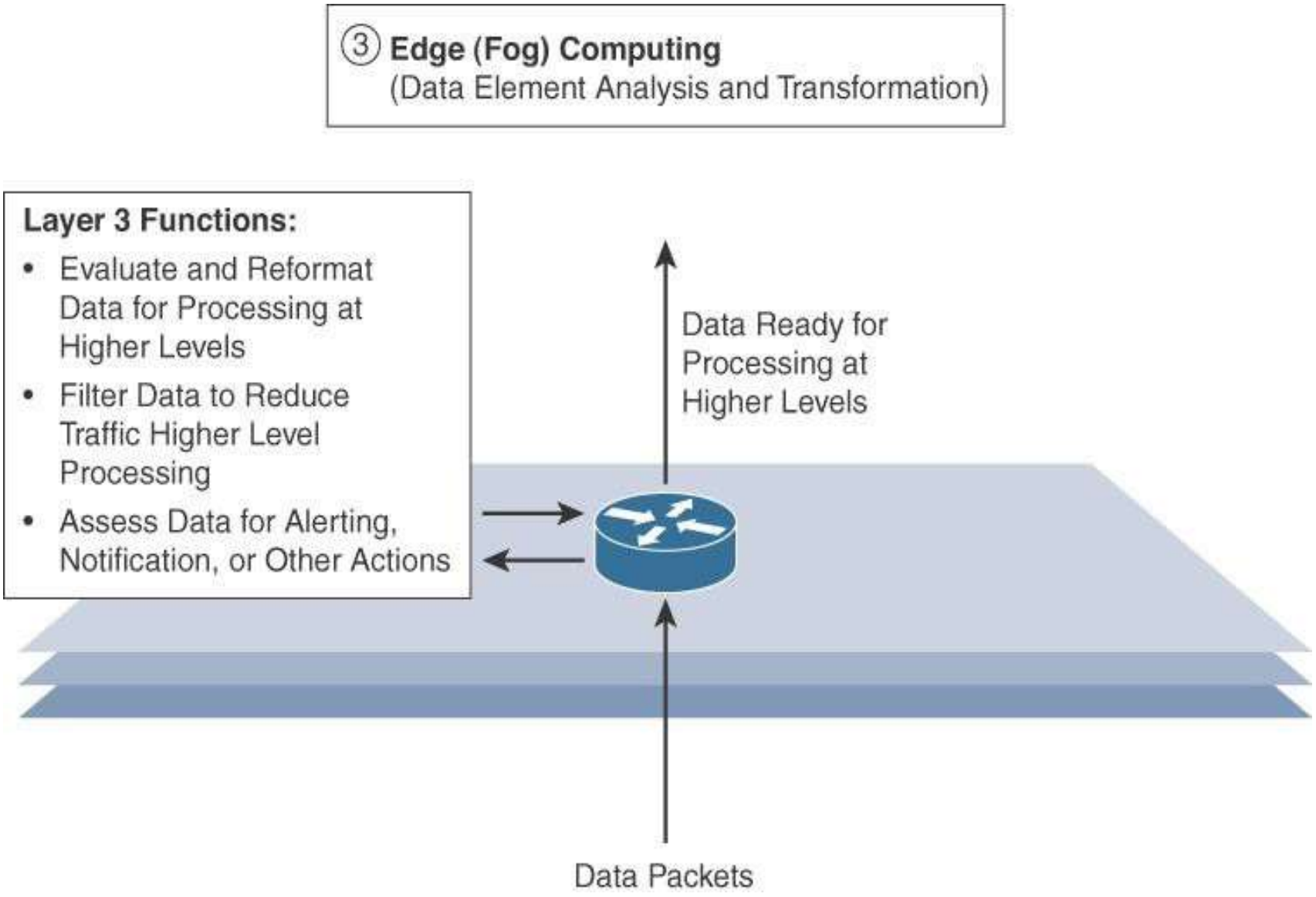
② **Connectivity**
(Communication and Processing Units)

Layer 2 Functions:

- Communications Between Layer 1 Devices
- Reliable Delivery of Information Across the Network
- Switching and Routing
- Translation Between Protocols
- Network Level Security

Layer 3: Edge Computing Layer

- Fog layer
- At this layer, the emphasis is on data reduction and converting network data flows into information that is ready for storage and processing by higher layers
- One of the basic principles of this reference model is that information processing is initiated as early and as close to the edge of the network as possible
- Another important function that occurs at Layer 3 is the evaluation of data to see if it can be filtered or aggregated before being sent to a higher layer
- This also allows for data to be reformatted or decoded, making additional processing by other systems easier

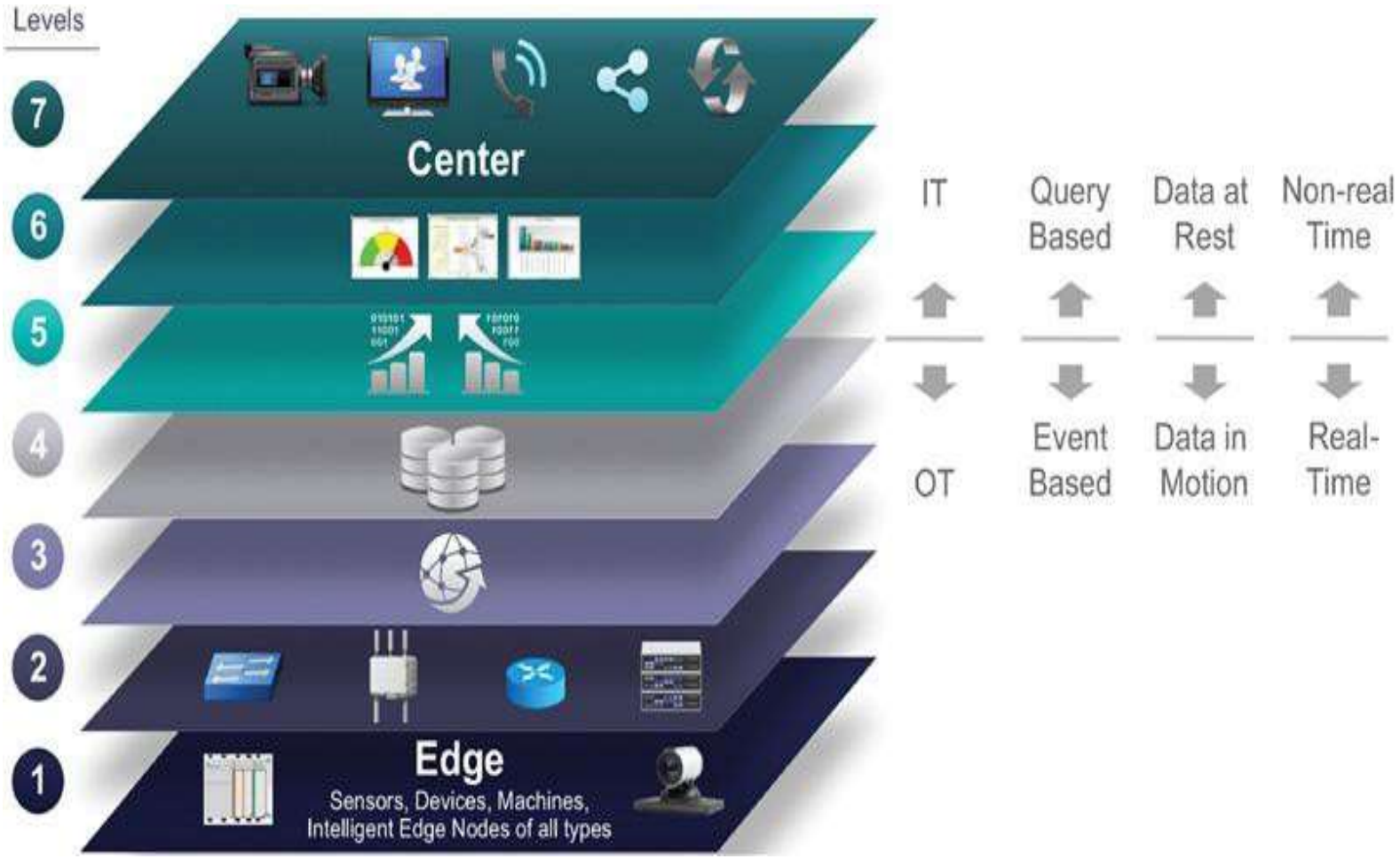

③ Edge (Fog) Computing
(Data Element Analysis and Transformation)
Layer 3 Functions:
• Evaluate and Reformat Data for Processing at Higher Levels
• Filter Data to Reduce Traffic Higher Level Processing
• Assess Data for Alerting, Notification, or Other Actions
Data Ready for Processing at Higher Levels
Data Packets

Upper Layers: Layers 4–7

- The upper layers deal with handling and processing the IoT data generated by the bottom layer

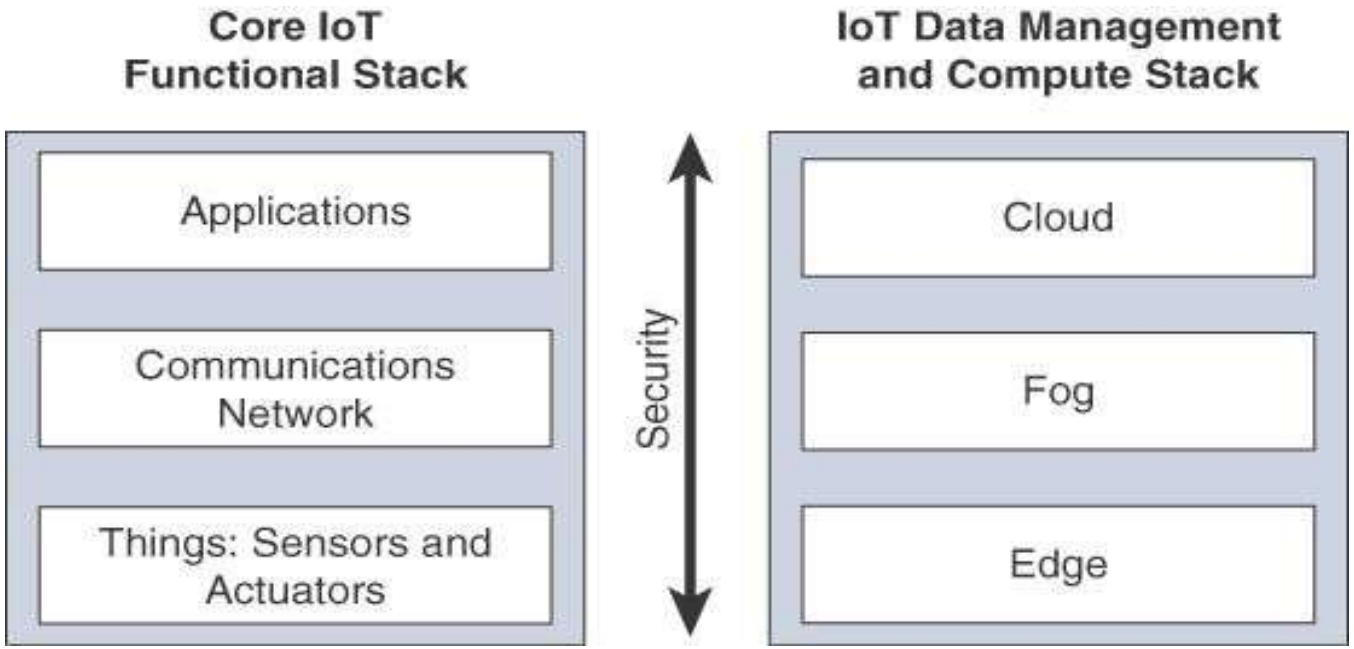
IoT Reference Model Layer	Functions
Layer 4: Data accumulation layer	Captures data and stores it so it is usable by applications when necessary. Converts event-based data to query-based processing.
Layer 5: Data abstraction layer	Reconciles multiple data formats and ensures consistent semantics from various sources. Confirms that the data set is complete and consolidates data into one place or multiple data stores using virtualization.
Layer 6: Applications layer	Interprets data using software applications. Applications may monitor, control, and provide reports based on the analysis of the data.
Layer 7: Collaboration and processes layer	Consumes and shares the application information. Collaborating on and communicating IoT information often requires multiple steps, and it is what makes IoT useful. This layer can change business processes and delivers the benefits of IoT.

IT and OT Responsibilities in the IoT Reference Model



A SIMPLIFIED IOT ARCHITECTURE

- The framework is presented as two parallel stacks:
- The IoT Data Management and Compute Stack and
- The Core IoT Functional Stack.



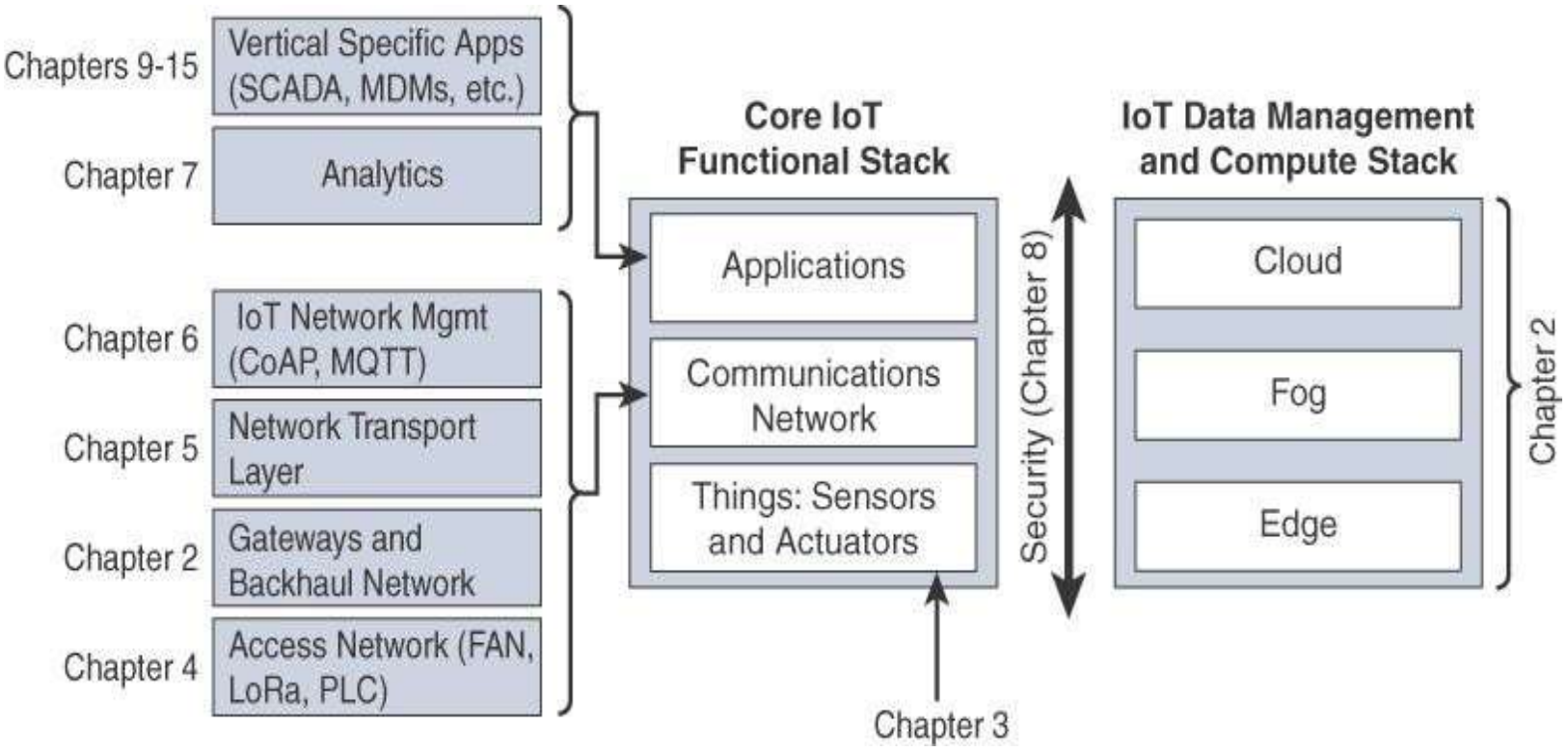
- The network communications layer of the IoT stack itself involves a significant amount of detail and incorporates a vast array of technologies.
- Consider for a moment the heterogeneity of IoT sensors and the many different ways that exist to connect them to a network.
- The network communications layer needs to consolidate these together, offer gateway and backhaul technologies, and ultimately bring the data back to a central location for analysis and processing.

- The network between the gateway and the data center is composed mostly of traditional technologies that experienced IT professionals would quickly recognize.
- These include tunneling and VPN technologies, IP-based quality of service (QoS), conventional Layer 3 routing protocols such as BGP and IP-PIM, and security capabilities such as encryption, access control lists (ACLs), and firewalls

- Unlike with most IT networks, the applications and analytics layer of IoT doesn't necessarily exist only in the data center or in the cloud.
- Due to the unique challenges and requirements of IoT, it is often necessary to deploy applications and data management throughout the architecture in a tiered approach, allowing data collection, analytics, and intelligent controls at multiple points in the IoT system

- The three data management layers are
- The edge layer (data management within the sensors themselves),
- The fog layer (data management in the gateways and transit network), and
- The cloud layer (data management in the cloud or central data center).

Expanded View of the Simplified IoT Architecture



THE CORE IOT FUNCTIONAL STACK

- IoT networks are built around the concept of "things," or smart objects performing functions and delivering new connected services.
- These objects are "smart" because they use a combination of contextual information and configured goals to perform actions
- These actions can be self-contained (that is, the smart object does not rely on external systems for its actions); however, in most cases, the "thing" interacts with an external system to report information that the smart object collects, to exchange with other objects, or to interact with a management platform

- From an architectural standpoint, several components have to work together for an IoT network to be operational:
- **“Things” layer:**
- At this layer, the physical devices need to fit the constraints of the environment in which they are deployed while still being able to provide the information needed.
- **Communications network layer:**
- When smart objects are not self-contained, they need to communicate with an external system.
- In many cases, this communication uses a wireless technology.
- This layer has four sublayers:

- **Access network sub layer:**
- This is typically made up of wireless technologies such as 802.11ah, 802.15.4g, and LoRa.
- The sensors connected to the access network may also be wired.

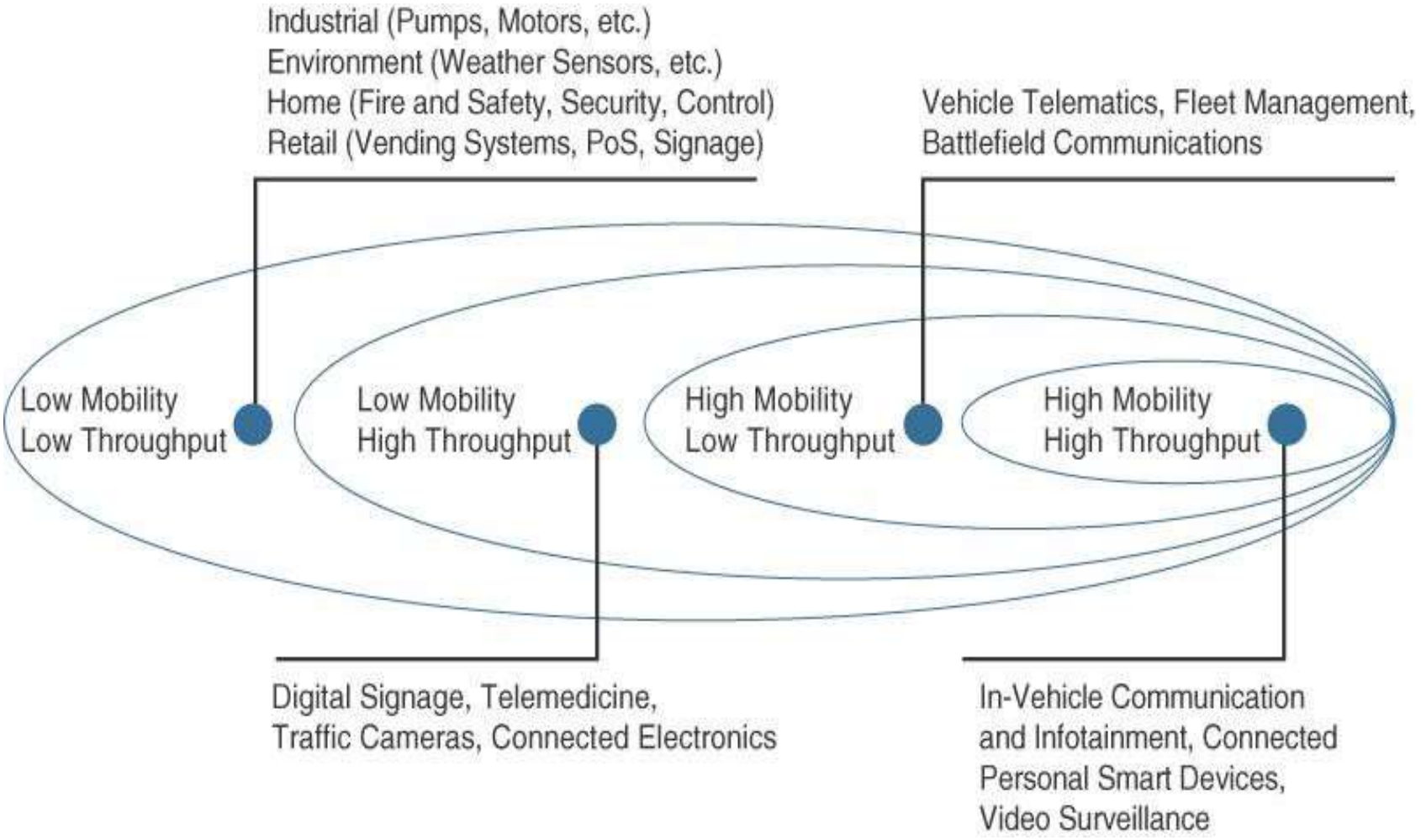
- **Gateways and backhaul network sub layer:**
- The gateway communicates directly with the smart objects.
- The role of the gateway is to forward the collected information through a longer-range medium (called the backhaul) to a headend central station where the information is processed
- **Network transport sub layer:**
- For communication to be successful, network and transport layer protocols such as IP and UDP must be implemented to support the variety of devices to connect and media to use

- **IoT network management sub layer:**
- Additional protocols must be in place to allow the headend applications to exchange data with the sensors.
- Examples include CoAP and MQTT.

- **Application and analytics layer:**
- At the upper layer, an application needs to process the collected data,
- To control the smart objects when necessary,
- To make intelligent decision based on the information collected and, in turn, instruct the "things" or other systems to adapt to the analyzed conditions and change their behaviors or parameters.

Layer 1: Things: Sensors and Actuators Layer

- Battery-powered or power-connected
- Mobile or static
- Low or high reporting frequency
- Simple or rich data
- Report range
- Object density per cell


Industrial (Pumps, Motors, etc.)
Environment (Weather Sensors, etc.)
Home (Fire and Safety, Security, Control)
Retail (Vending Systems, PoS, Signage)
Vehicle Telematics, Fleet Management,
Battlefield Communications
Low Mobility
Low Throughput
Low Mobility
High Throughput
High Mobility
Low Throughput
High Mobility
High Throughput
Digital Signage, Telemedicine,
Traffic Cameras, Connected Electronics
In-Vehicle Communication
and Infotainment, Connected
Personal Smart Devices,
Video Surveillance

Layer 3: Applications and Analytics Layer

- Once connected to a network, your smart objects exchange information with other systems
- **Analytics Versus Control Applications**
- **Analytics application:**
- This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed
- The important aspect is that the application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object.

- **Control application:**
- This type of application controls the behavior of the smart object or the behavior of an object related to the smart object.
- For example, a pressure sensor may be connected to a pump. A control application increases the pump speed when the connected sensor detects a drop in pressure.
- Control applications are very useful for controlling complex aspects of an IoT network with a logic that cannot be programmed inside a single IoT object

- **DataVersus Network Analytics**
- **Data analytics:**
- This type of analytics processes the data collected by smart objects and combines it to provide an intelligent view related to the IoT system.
- At a very basic level, a dashboard can display an alarm when a weight sensor detects that a shelf is empty in a store
- In a more complex case, temperature, pressure, wind, humidity, and light levels collected from thousands of sensors may be combined and then processed to determine the likelihood of a storm and its possible path

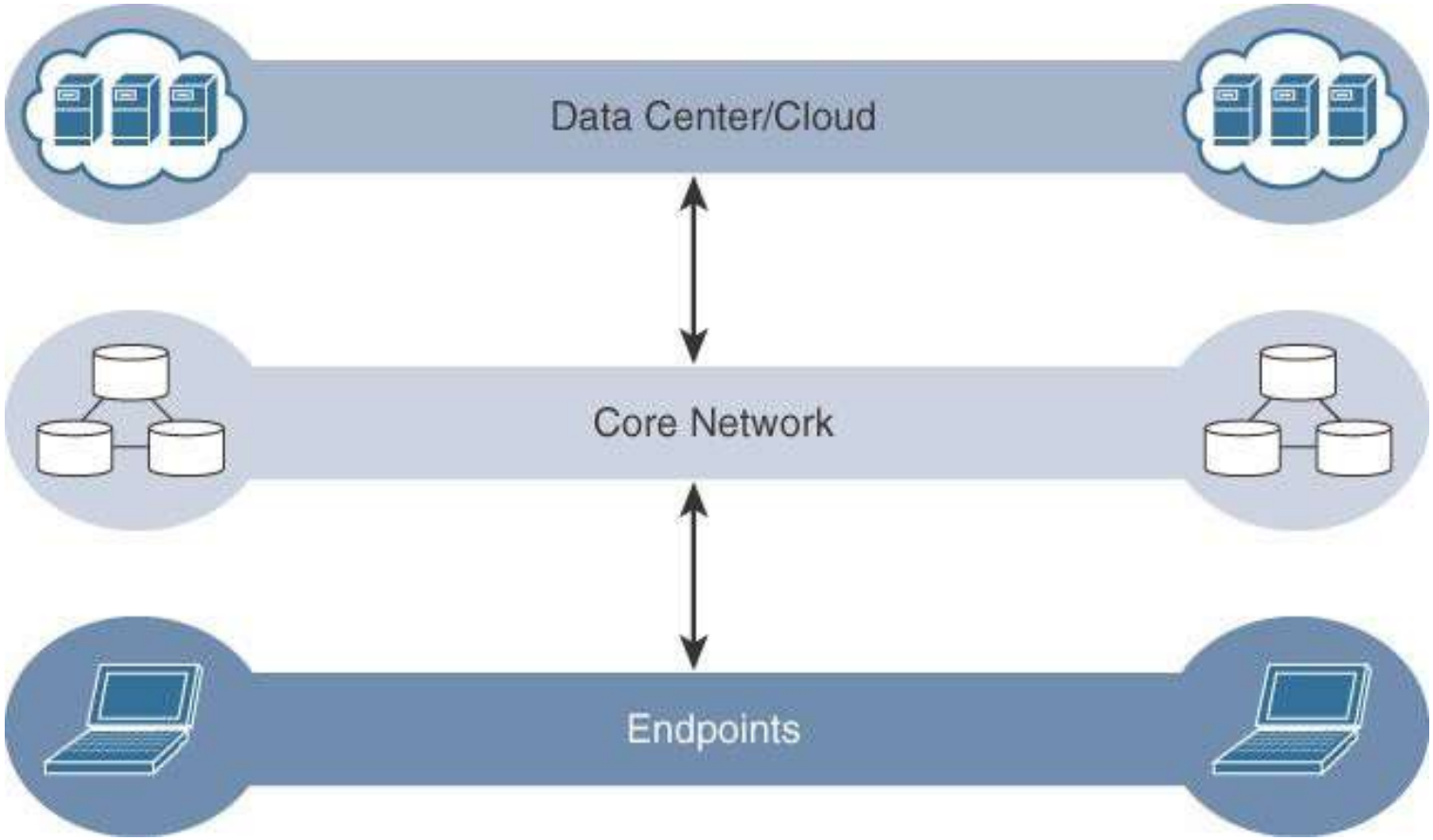
- **Network analytics:**
- Most IoT systems are built around smart objects connected to the network.
- A loss or degradation in connectivity is likely to affect the efficiency of the system. Such a loss can have dramatic effects.

IOT DATA MANAGEMENT AND COMPUTE STACK

- The data generated by IoT sensors is one of the single biggest challenges in building an IoT system
- In sensor networks, the vast majority of data generated is unstructured and of very little use on its own
- In most cases, the processing location is outside the smart object.
- A natural location for this processing activity is the cloud.
- Smart objects need to connect to the cloud, and data processing is centralized
- One advantage of this model is simplicity

- Limitations
- As data volume, the variety of objects connecting to the network, and the need for more efficiency increase, new requirements appear, and those requirements tend to bring the need for data analysis closer to the IoT system
- Minimizing latency
- Conserving network bandwidth
- Increasing local efficiency

Data management in traditional IT systems



IoT issues to be addressed

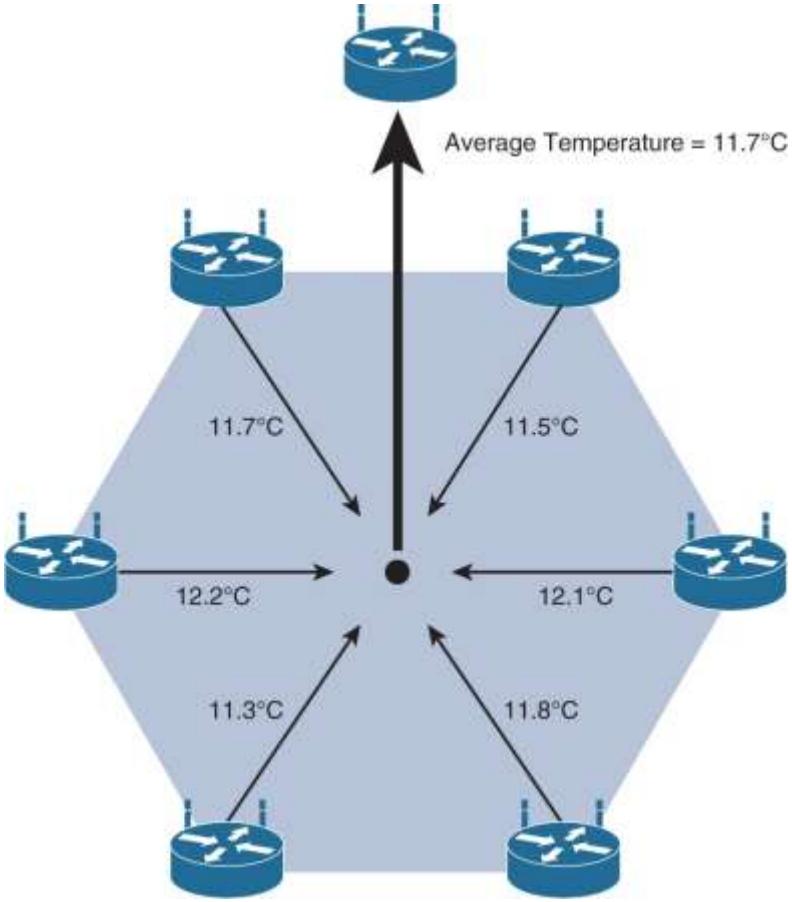
- Bandwidth in last-mile IoT networks is very limited
- Latency can be very high
- Network backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links
- The volume of data transmitted over the backhaul can be high
- Big data is getting bigger

Fog Computing

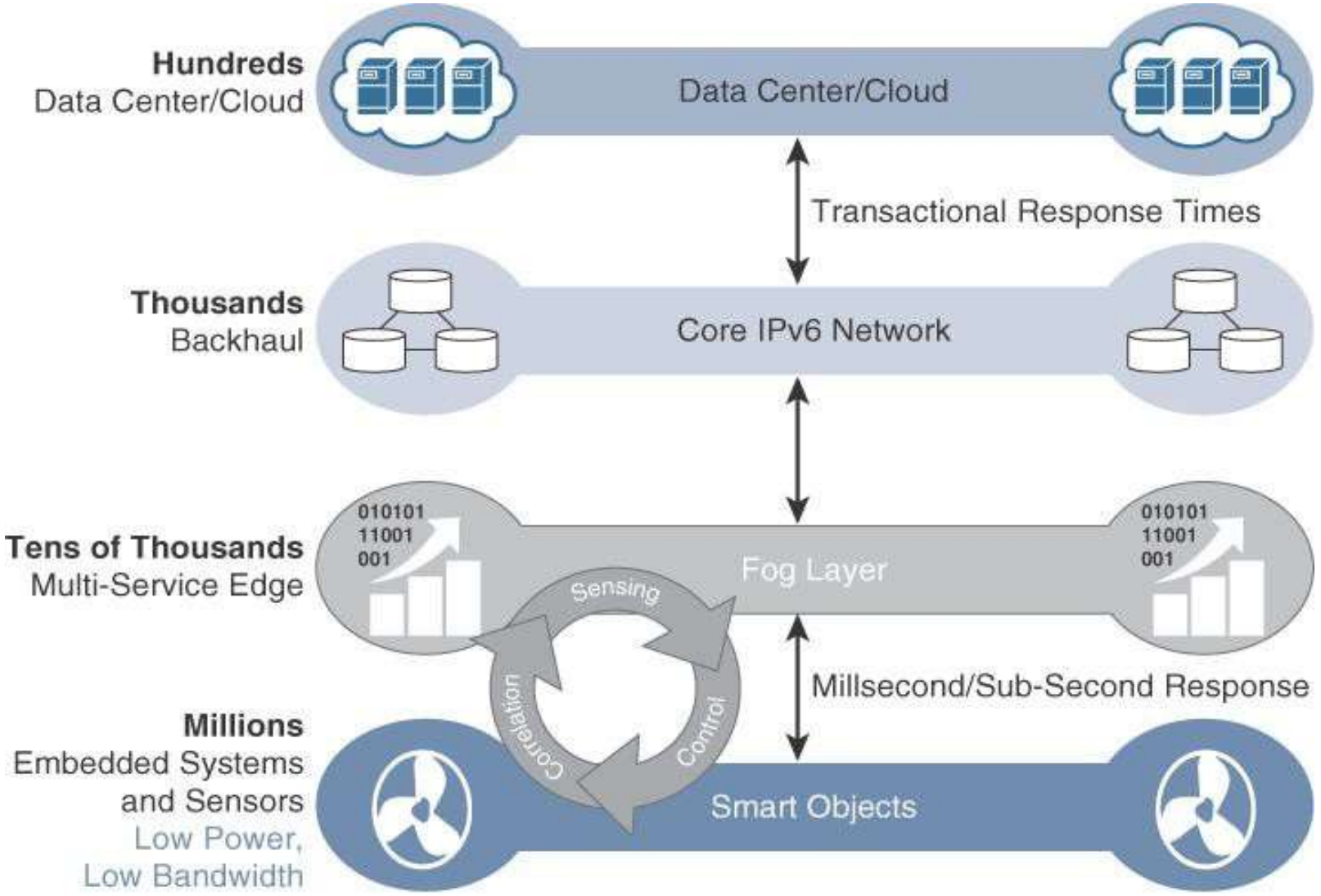
- The solution to the challenges mentioned in the previous section is to distribute data management throughout the IoT system, as close to the edge of the IP network as possible
- Any device with computing, storage, and network connectivity can be a fog node.
- Examples include industrial controllers, switches, routers, embedded servers, and IoT gateways.
- Analyzing IoT data close to where it is collected minimizes latency, offloads gigabytes of network traffic from the core network, and keeps sensitive data inside the local network.

- An advantage of this structure is that the fog node allows intelligence gathering (such as analytics) and control from the closest possible point, and in doing so, it allows better performance over constrained networks

Illustration of Edge/Fog Computing: Data Aggregation in Wireless Sensor Networks



The IoT Data Management and Compute Stack with Fog Computing

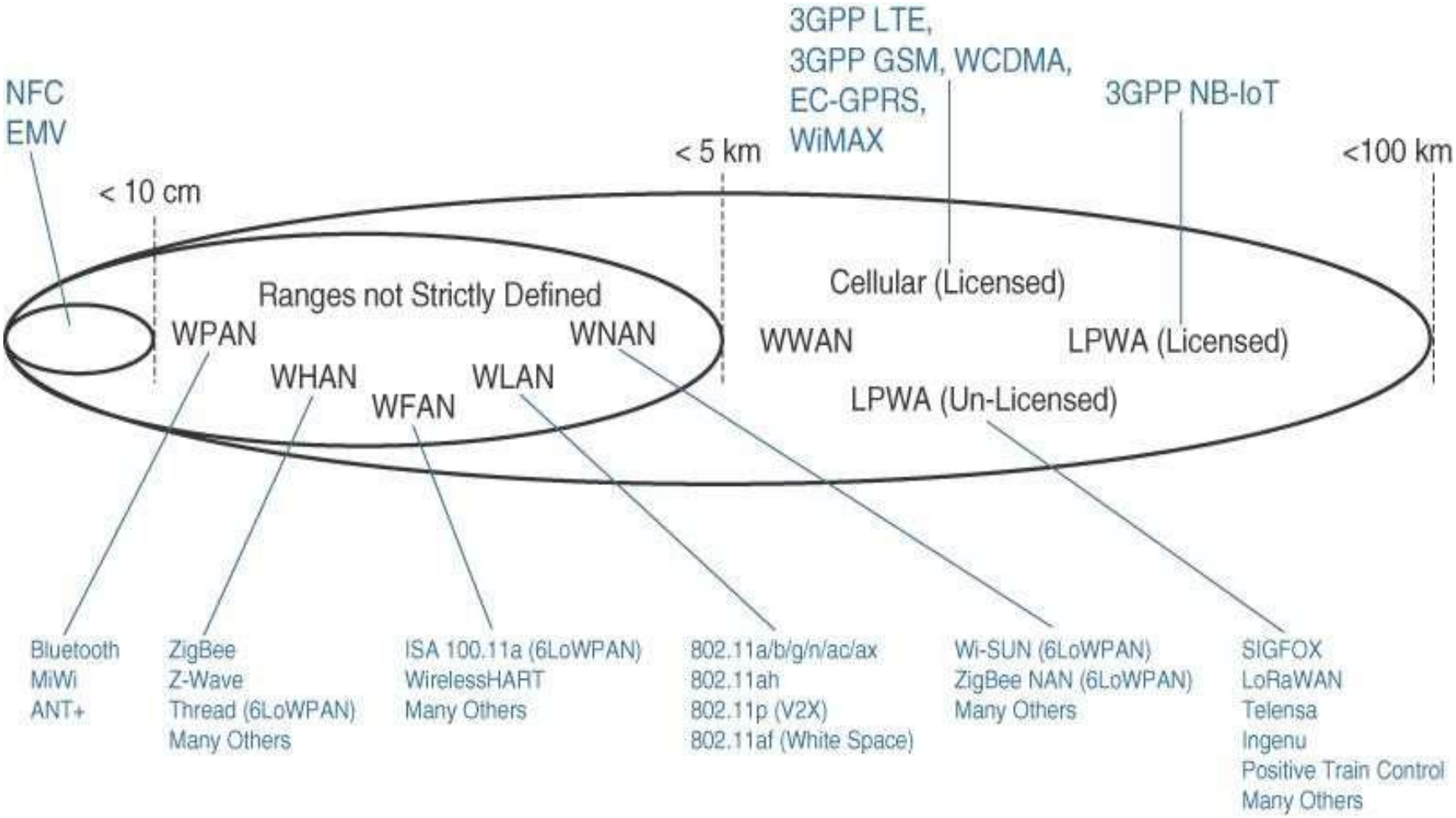


- The defining characteristic of fog computing are as follows:
  - **Contextual location awareness and low latency**
  - **Geographic distribution**
  - **Deployment near IoT endpoints**
  - **Wireless communication between the fog and the IoT endpoint**
  - **Use for real-time interactions**

Edge Computing

- Computing resides directly in the sensors and IoT devices.
- New classes of IoT endpoints have enough compute capabilities to perform at least low-level analytics and filtering to make basic decisions

Access Technologies and Distances



WPAN: Wireless Personal Area Network
WHAN: Wireless Home Area Network
WFAN: Wireless Field (or Factory) Area Network
WLAN: Wireless Local Area Network

WNAN: Wireless Neighborhood Area Network
WWAN: Wireless Wide Area Network
LPWA: Low Power Wide Area

PAN (personal area network):

- Scale of a few meters.
- This is the personal space around a person.
- A common wireless technology for this scale is Bluetooth.

HAN (home area network):

- Scale of a few tens of meters.
- At this scale, common wireless technologies for IoT include ZigBee and Bluetooth Low Energy (BLE).

NAN (neighborhood area network):

- Scale of a few hundreds of meters.
- The term NAN is often used to refer to a group of house units from which data is collected.

FAN (field area network):

- **Scale of several tens of meters to several** hundred meters.
- FAN typically refers to an outdoor area larger than a single group of house units.
- A FAN is sometimes viewed as a group of NANs

- **LAN (local area network):**
- Scale of up to 100 m.
- MAN,WAN are also used

- Increasing the throughput and achievable distance typically comes with an increase in power consumption
- Therefore, after determining the smart object requirements (in terms of mobility and data transfer), a second step is to determine the target quantity of objects in a single collection cell, based on the transmission range and throughput required.This parameter in turn determines the size of the cell.

- It may be tempting to simply choose the technology with the longest range and highest throughput.
- However, the cost of the technology is a third determining factor
- The amount of data to carry over a given time period along with correlated power consumption (driving possible limitations in mobility and range) determines the wireless cell size and structure

- **Point-to-point topologies:**
- These topologies allow one point to communicate with another point
- **Point-to-multipoint topologies:**
- These topologies allow one point to communicate with more than one other point.
- Most IoT technologies where one or more than one gateways communicate with multiple smart objects are in this category

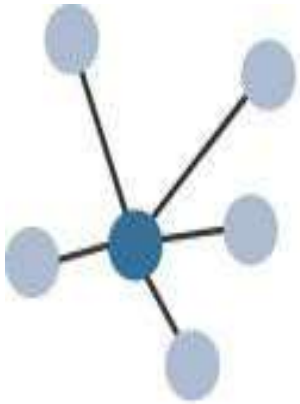
- To form a network, a device needs to connect with another device.
- When both devices fully implement the protocol stack functions, they can form a peer-to-peer network
- However, in many cases, one of the devices collects data from the others.
- For example, in a house, temperature sensors may be deployed in each room or each zone of the house, and they may communicate with a central point where temperature is displayed and controlled.
- A room sensor does not need to communicate with another room sensor. In that case, the control point is at the center of the network.
- The network forms a star topology

- In such a configuration, the central point can be in charge of the overall network coordination, taking care of the beacon transmissions and connection to each sensor.
- In the IEEE 802.15.4 standard, the central point is called a *coordinator* for the network.
- With this type of deployment, each sensor is not intended to do anything other than communicate with the coordinator in a master/slave type of relationship.

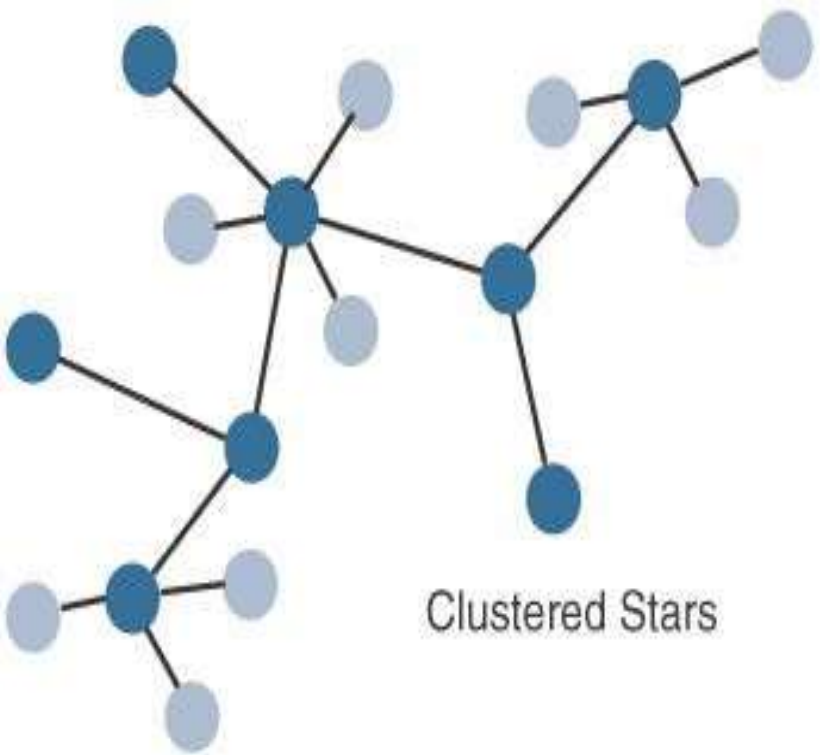
- The sensor can implement a subset of protocol functions to perform just a specialized part (communication with the coordinator).
- Such a device is called a reduced-function device (RFD).
- An RFD cannot be a coordinator.
- An RFD also cannot implement direct communications to another RFD.

- The coordinator that implements the full network functions is called, by contrast, a full-function device (FFD).
- An FFD can communicate directly with another FFD or with more than one FFD, forming multiple peer-to-peer connections.
- Topologies where each FFD has a unique path to another FFD are called cluster tree topologies.
- FFDs in the cluster tree may have RFDs, resulting in a cluster star topology.

Star and Clustered Star Topologies



Star Topology



Clustered Stars

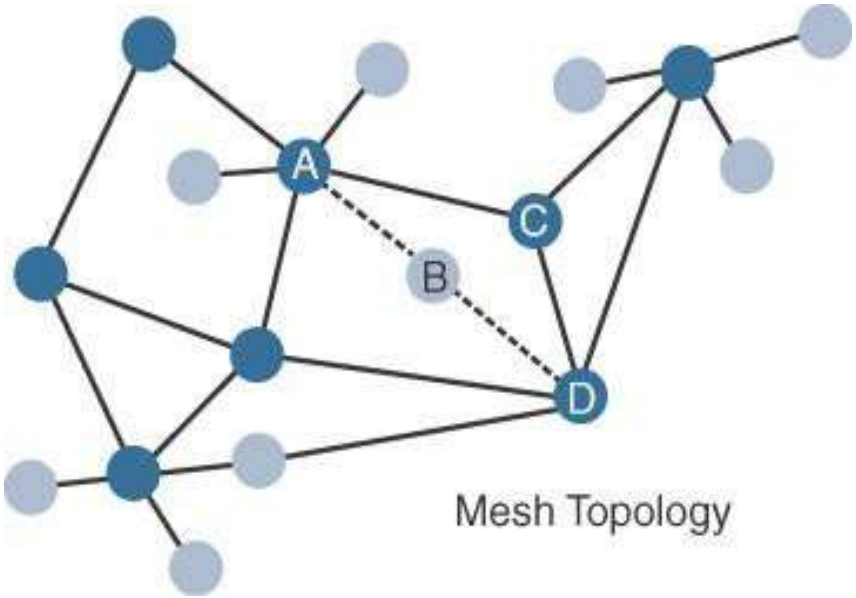
Full Function Device

Reduced Function Device

- Other point-to-multipoint technologies allow a node to have more than one path to another node, forming a mesh topology.
- This redundancy means that each node can communicate with more than just one other node.
- This communication can be used to directly exchange information between nodes (the receiver directly consumes the information received) or to extend the range of the communication link.
- In this case, an intermediate node acts as a relay between two other nodes

- Another property of mesh networks is redundancy.
- The disappearance of one node does not necessarily interrupt network communications.
- Data may still be relayed through other nodes to reach the intended destination

Mesh Topology



Mesh Topology

Gateways and Backhaul Sublayer

- Data collected from a smart object may need to be forwarded to a central station where data is processed.
- As this station is often in a different location from the smart object, data directly received from the sensor through an access technology needs to be forwarded to another medium (the backhaul) and transported to the central station.
- The gateway is in charge of this inter-medium communication
- In most cases, the smart objects are static or mobile within a limited area.
- The gateway is often static
- However, some IoT technologies do not apply this model

- For example, dedicated short-range communication (DSRC) allows vehicle-to-vehicle and vehicle-to-infrastructure communication
- A wireless technology is used for backhaul communication, peer-to-peer, or mesh communication between vehicles
- The choice of a backhaul technology depends on the communication distance and also on the amount of data that needs to be forwarded.
- When the smart object's operation is controlled from a local site, and when the environment is stable (for example, factory or oil and gas field), Ethernet can be used as a backhaul

- In unstable or changing environments (for example, open mines) where cables cannot safely be run, a wireless technology is used.
- Wi-Fi is common in this case, often with multiple hops between the sensor field and the operation center.
- Mesh is a common topology to allow communication flexibility in this type of dynamic environment
- However, throughput decreases as node-to-node distance increases, and it also decreases as the number of hops increases

Architectural Considerations for WiMAX and Cellular Technologies

Technology	Type and Range	Architectural Characteristics
Ethernet	Wired, 100 m max	Requires a cable per sensor/sensor group; adapted to static sensor position in a stable environment; range is limited; link is very reliable
Wi-Fi (2.4 GHz, 5 GHz)	Wireless, 100 m (multipoint) to a few kilometers (P2P)	Can connect multiple clients (typically fewer than 200) to a single AP; range is limited; adapted to cases where client power is not an issue (continuous power or client battery recharged easily); large bandwidth available, but interference from other systems likely; AP needs a cable
802.11ah (HaloW, Wi-Fi in sub-1 GHz)	Wireless, 1.5 km (multipoint), 10 km (P2P)	Can connect a large number of clients (up to 6000 per AP); longer range than traditional Wi-Fi; power efficient; limited bandwidth; low adoption; and cost may be an issue
WiMAX (802.16)	Wireless, several kilometers (last mile), up to 50 km (backhaul)	Can connect a large number of clients; large bandwidth available in licensed spectrum (fee-based); reduced bandwidth in license-free spectrum (interferences from other systems likely); adoption varies on location
Cellular (for example, LTE)	Wireless, several kilometers	Can connect a large number of clients; large bandwidth available; licensed spectrum (interference-free; license-based)

Network Transport Sublayer

- The practical implementations are often flexible, with multiple transversal communication paths
- For example, consider the case of IoT for the energy grid.
- Your house may have a meter that reports the energy consumption to a gateway over a wireless technology.
- Other houses in your neighborhood (NAN) make the same report, likely to one or several gateways
- The data to be transported is small and the interval is large (for example, four times per hour), resulting in a low-mobility, low-throughput type of data structure, with transmission distances up to a mile
- Several technologies (such as 802.11ah, 802.15.4, or LPWA) can be used for this collection segment. Other neighborhoods may also connect the same way, thus forming a FAN

IoT Network Management Sublayer

- IP, TCP, and UDP bring connectivity to IoT networks.
- Upper-layer protocols need to take care of data transmission between the smart objects and other systems
- Multiple protocols have been leveraged or created to solve IoT data communication problems.
- Some networks rely on a push model (that is, a sensor reports at a regular interval or based on a local trigger), whereas others rely on a pull model (that is, an application queries the sensor over the network), and multiple hybrid approaches are also possible

- Some IoT implementers have suggested HTTP for the data transfer phase.
- The sensor could use the client part to establish a connection to the IoT central application (the server), and then data can be exchanged
- But HTTP was not designed to operate in constrained environments with low memory, low power, low bandwidth, and a high rate of packet failure
- One example isWebSocket.
- WebSocket is part of the HTML5 specification, and provides a simple bidirectional connection over a single connection.
- Some IoT solutions use WebSocket to manage the connection between the smart object and an external application

- Extensible Messaging and Presence Protocol (XMPP)
- XMPP is based on instant messaging and presence.
- It allows the exchange of data between two or more systems and supports presence and contact list maintenance.
- It can also handle publish/subscribe, making it a good choice for distribution of information to multiple devices

- Constrained Application Protocol (CoAP)
- CoAP uses some methods similar to those of HTTP (such as Get, Post, Put, and Delete) but implements a shorter list, thus limiting the size of the header.
- CoAP also runs on UDP (whereas HTTP typically uses TCP).
- CoAP also adds a feature that is lacking in HTTP and very useful for IoT: observation.
- Observation allows the streaming of state changes as they occur, without requiring the receiver to query for these changes

- Message Queue Telemetry Transport (MQTT)
- MQTT uses a broker-based architecture.
- The sensor can be set to be an MQTT publisher (publishes a piece of information), the application that needs to receive the information can be set as the MQTT subscriber, and any intermediary system can be set as a broker to relay the information between the publisher and the subscriber(s).
- MQTT runs over TCP

Layer 3: Applications and Analytics Layer

- Once connected to a network, your smart objects exchange information with other systems
- **Analytics Versus Control Applications**
- **Analytics application:**
- This type of application collects data from multiple smart objects, processes the collected data, and displays information resulting from the data that was processed
- The important aspect is that the application processes the data to convey a view of the network that cannot be obtained from solely looking at the information displayed by a single smart object

The Hierarchy of Edge, Fog, and Cloud

- It is important to stress that edge or fog computing in no way replaces the cloud
- This model suggests a hierarchical organization of network, compute, and data storage resources.
- At each stage, data is collected, analyzed, and responded to when necessary, according to the capabilities of the resources at each layer.
- As data needs to be sent to the cloud, the latency becomes higher.
- The advantage of this hierarchy is that a response to events from resources close to the end device is fast and can result in immediate benefits, while still having deeper compute resources available in the cloud when necessary.

- It is important to note that the heterogeneity of IoT devices also means a heterogeneity of edge and fog computing resources.
- While cloud resources are expected to be homogenous, it is fair to expect that in many cases both edge and fog resources will use different operating systems, have different CPU and data storage capabilities, and have different energy consumption profiles.

- The most time-sensitive data is analyzed on the edge or fog node closest to the things generating the data.
- Data that can wait seconds or minutes for action is passed along to an aggregation node for analysis and action.
- Data that is less time sensitive is sent to the cloud for historical analysis, big data analytics, and long-term storage

- In summary, when architecting an IoT network, you should consider the amount of data to be analyzed and the time sensitivity of this data.
- Understanding these factors will help you decide whether cloud computing is enough or whether edge or fog computing would improve your system efficiency.
- Fog computing accelerates awareness and response to events by eliminating a round trip to the cloud for analysis
- It avoids the need for costly bandwidth additions by offloading gigabytes of network traffic from the core network.
- It also protects sensitive IoT data by analyzing it inside company walls.